

第 2 章 8086/8088 微处理器

内容提要:

8086 是 X86 架构的鼻祖。本章介绍微机系统的核心部件——微处理器(CPU)。对具有典型基础意义的 8086/8088 微处理器详细解析了其功能结构、寄存器结构、存储器寻址方法、各引脚功能以及总线时序。

教学目标:

- 掌握微处理器的功能结构组成。
- 理解和掌握微处理器的寄存器结构,明确各寄存器的用法。
- 理解“段加偏移”的存储器寻址方法,掌握物理地址和逻辑地址的关系。
- 掌握微处理器的总线周期的概念,理解微机系统在时序作用下的工作过程。

2.1 8086/8088 微处理器特点

20 世纪 80 年代初,IBM 公司用 Intel 8088 作为中央处理器(CPU),推出了个人计算机 IBM-PC/XT 系统(现称微机系统或 PC),从此,开创了个人计算机系统的先河,微处理器 8086/8088 成为微机系统的典型芯片。

在 PC 系列微机中,应用最广泛的微处理器是 Intel 公司的 X86 系列,为了保持产品的兼容性和维护老用户在软件上的投资不受损失,Intel 公司推出的新一代 X86 系列微处理器一定兼容 8086/8088 微处理器指令系统,这就决定了新型芯片是在老芯片基础上改进和发展起来的。因此,X86 系列(包括 Pentium 和 Pentium III)都是在 8086/8088 基础上逐步改进发展而来的。学习 8086/8088 微处理器的基本结构原理就成为掌握 X86 系列处理器的基础。

8088 微处理器是 IBM PC/XT 微型计算机系统板的核心部件。在这片大规模集成电路芯片上,集成了控制整个微型计算机工作的指令译码电路、微程序控制电路、寄存器组及算术逻辑运算部件(ALU)。它是一种通用的高性能的准 16 位微处理器。

8086 是 Intel 系列的 16 位微处理器。8086 和 8088 的内部结构基本相同,所不同的是:8086 的外部数据总线为 16 位,而 8088 的外部数据总线为 8 位,因此,称 8086 为 16 位微处理器,而 8088 为准 16 位微处理器。功能强大的微处理器 8086/8088 还有如下一些特点:

- (1) 指令系统功能齐全,各类指令共 100 多条。
- (2) 多种寻址方式,适用于高级语言中的数组和记录等数据结构形式。
- (3) 20 位地址线。存储器寻址范围可达 1MB 的存储空间。
- (4) 16 位 I/O 端口地址线,I/O 接口可寻址 64K 端口地址。
- (5) 具有较强中断处理能力。能处理软件中断、非屏蔽中断和可屏蔽中断三类中断源。
- (6) 具有管理和响应 DMA 操作的能力。
- (7) 可实现多处理器协调和管理总线的能力。

2.2 8086/8088 微处理器结构

了解 8086/8088 CPU 内部结构是理解微机工作原理的重要基础，其寄存器构成及作用是编写汇编语言程序所必须掌握的。掌握本节内容对后续章节的学习非常重要。

2.2.1 8086/8088 功能结构

在 8086/8088 之前，微处理器执行指令的过程是串行的，即取指令后分析执行，继而取下一条指令再分析执行。为了使取指令与分析、执行指令能并行进行，提高 CPU 的执行效率，8086/8088 CPU 由两大模块，即总线接口部件 (Bus Interface Unit, BIU) 和执行部件 (Execution Unit, EU) 组成，如图 2.2-1 所示。

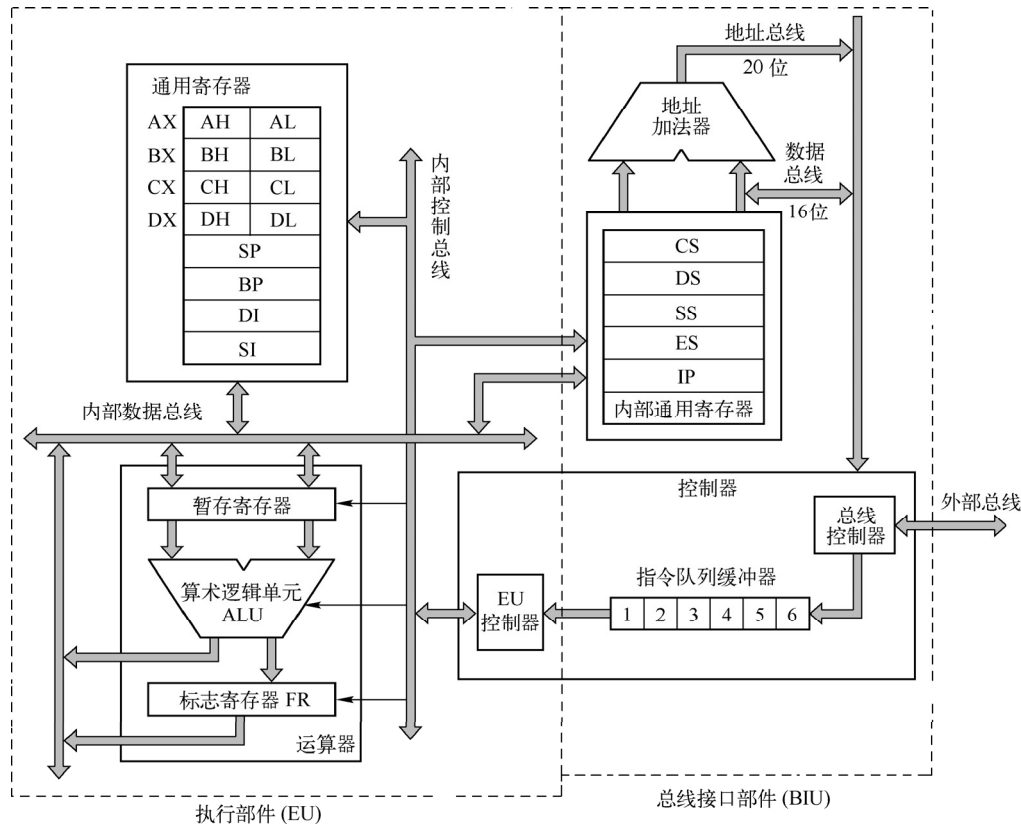


图 2.2-1 8086 内部功能结构

1. 总线接口部件 (BIU)

总线接口部件 (BIU) 负责在 CPU 与存储器、I/O 端口之间传送数据。BIU 完成下述功能：

- (1) 保持当前要执行的指令、分析指令、向执行部件提供稳定的指令特征状态；
- (2) 计算下条指令的地址，以便控制程序的走向，保证正确地执行程序；
- (3) 控制存储器、输入/输出设备之间的数据交换。

BIU 的内部结构由 4 个段寄存器 (CS、DS、SS、ES)、指令指针寄存器 IP、完成与 EU 通信的内部寄存器、地址加法器和指令队列缓冲器组成。这些寄存器的主要功能如下：

- (1) 地址加法器将段地址寄存器的内容和地址偏移量相加，生成 20 位物理地址。
- (2) 16 位指令指针寄存器 (IP)，类似于前面介绍的程序计数器 (PC)，它可用于以下 3 种

- 情况：
- ① 当取指令时，用代码段寄存器 CS 加上 IP 偏移量，形成要取出的指令字节的 20 位物理地址。每取一个指令操作码字节后，IP 自动加 1。
 - ② 形成指令队列时，按 IP 顺序取出的指令字节装入指令队列中，进行排队，等待执行，程序员不能对 IP 进行存取操作。
 - ③ 执行转移类指令、子程序调用和返回指令及中断响应时，能按照指令自动修改 IP 的内容。
- BIU 部件提供存储器与输入/输出接口的地址、数据和控制信号线。图 2.2-1 所示的外部总线是这些信号线的总称，后面会详细介绍具体引脚。

2. 执行部件(EU)

执行部件(EU)负责指令的执行。EU 由算术逻辑单元、暂存寄存器、标志寄存器、通用寄存器组和 EU 控制器构成。其主要任务是执行指令，进行全部算术逻辑运算，完成偏移地址的计算，向总线接口单元(BIU)提供指令执行结果的数据和偏移地址，并对通用寄存器和标志寄存器进行管理。

取指令部分与执行指令部分是分开的，在一条指令的执行过程中，就可以读取下一条及后续的多条指令到指令流队列中排队。这种重叠式的操作技术，过去只在大型计算机中才使用。

3. 指令队列缓冲器

指令队列缓冲器是暂存指令的寄存器，又称为指令栈。它由 6 个(8088 是 4 个)八位的寄存器组成。8086/8088 在指令译码和执行指令的同时，总线接口部件从存储器中取下面一条或几条指令，取来的指令就放在指令队列中。一般情况下，CPU 执行完一条指令就可以立即执行下一条指令，从而提高了 CPU 执行指令的速度。

如图 2.2-2 所示，指令队列缓冲器遵循先装入的指令先取出、后装入的后取出。开始时队列缓冲器是空的，执行部件处于等待状态，取出的第 1 条指令放入队列缓冲器，当执行部件执行第 1 条指令时，便从队列缓冲器取走了第 1 条指令，同时 BIU 又取出第 2 条指令，并存入指令队列缓冲器。由于执行部件第 1 条指令尚未执行完，队列缓冲器未滿，于是总线接口部件又开始取第 3 条指令，并存入指令队列缓冲器。执行部件执行完第 1 条指令后，从队列缓冲器中又取走第 2 条指令执行。在执行第 2 条指令期间，总线接口部件又接连取出第 4、第 5 条指令存入队列缓冲器。执行部件执行第 2 条指令时需要取出操作数，于是总线接口部件重叠地处理，什么时候停止重叠取指令，什么时候清除队列缓冲器，什么时间开始取指令，主要视队列缓冲器是否空或滿，如果符合下列情况，便开始操作：

- (1) 只要指令队列缓冲器中有 2 个或 2 个以上(8088 是 1 个)字节未装指令时，便自动执行取指令操作；
- (2) 如果指令队列已填满指令时，便自动停止取指令操作；
- (3) 如果执行部件执行完转移、调用和返回指令时，则要清除指令队列缓冲器。

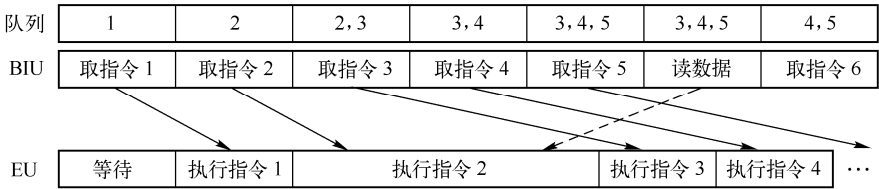


图 2.2-2 重叠执行指令的过程

2.2.2 8086/8088 寄存器结构

8086/8088 CPU 内部具有 14 个 16 位寄存器，形成一个寄存器组，用于提供运算中需要的操作数及中间结果，控制指令执行和对指令及操作数寻址，其寄存器结构如图 2.2-3 所示。按寄存器功能可分为通用寄存器、段寄存器和控制寄存器几类。

1. 通用寄存器

如图 2.2-3 所示，通用寄存器分为数据寄存器、地址指针和变址寄存器。

(1) 通用数据寄存器组

通用数据寄存器组包括 4 个 16 位的寄存器：AX(累加器)、BX(基地址寄存器)、CX(计数器)和 DX(数据寄存器)。在指令执行的过程中，既可用来寄存操作数，也可用于寄存操作的结果。它们中每一个又分成独立的两个 8 位寄存器，分别对应高 8 位(AH, BH, CH 和 DH)与低 8 位(AL, BL, CL 和 DL)。8 位寄存器只能用来存放数据；16 位寄存器主要用来存放数据，也可用来存放地址。

(2) 地址指针和变址寄存器

地址指针和变址寄存器包括 SP、BP、SI 和 DI，也是 4 个 16 位的寄存器，这组寄存器在功能上的共同点是在对存储器操作数寻址时，用于形成 20 位物理地址码的组成部分。在任何情况下，它们都不能独立地形成访问内存的地址码，因为它们只有 16 位，访问存储器的地址码由段地址和段内偏移地址两部分构成，而这 4 个寄存器用于存放段内偏移地址的全部或一部分。

它们也可以用来存放一个 16 位数据，作为一般的 16 位寄存器使用。但更多的是用它们提供段内寻址的偏移地址。

① 堆栈指针寄存器(Stack Pointer, SP)：主要用于指示堆栈的栈顶位置，必须与段寄存器 SS 一起形成堆栈的顶部地址，进行堆栈操作。SP 始终指向栈顶位置。

② 基址指针寄存器(Base Pointer, BP)：用做堆栈的一个附加指针，与 SS 联用，确定堆栈中某一存储单元的物理地址，用于对栈区的数据进行操作。它与 SP 的区别为：它不具有 SP 始终指向栈顶位置的功能，但它可以作为栈区内的一个偏移地址，访问栈区内任意位置的存储单元。

③ 源变址寄存器(Source Index, SI)和目的变址寄存器(Destination Index, DI)：SI 和 DI 寄存器具有自动增量和自动减量的功能，因此常与 DS、ES 联用，用于数据区中的数据块或字符串传送操作。在这类操作指令中，SI 指示源地址，而 DI 指示目的地址。

2. 段寄存器

在 8086/8088 系统中，访问存储器的地址码由段地址和段内偏移地址两部分组成，段寄存器用来存放段地址。总线接口单元(BIU)设置 4 个段寄存器，CPU 可通过 4 个段寄存器访问存储器中 4 个不同的段(每段 64KB)。这 4 个段寄存器分别是：

① 代码段寄存器(CS)：用于存放程序段的地址，在取指令时用此寄存器的内容左移 4 位以后的值加上指令计数器(指令指针 IP)中的偏移值，结果就是当前要取的指令的物理地址。

② 数据段寄存器(DS)：它存放当前数据段地址。执行指令时，根据指令寻址方式中段的

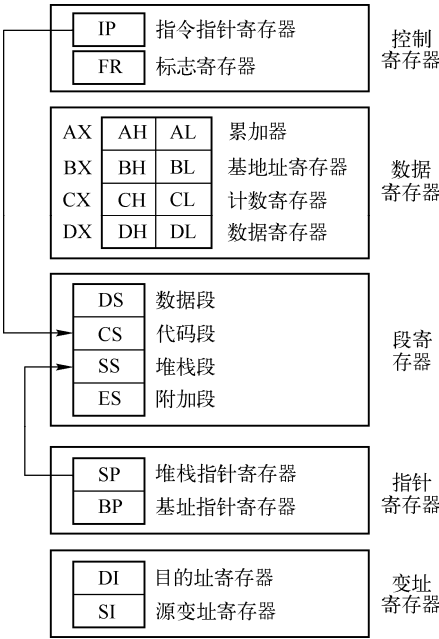


图 2.2-3 8086/8088 寄存器结构

指定，在执行部件中获得偏移地址，这一地址加上数据段寄存器左移 4 位后的值，结果就是实际要用的操作数的物理地址。

③ 堆栈段寄存器(SS)：栈操作时堆栈指针寄存器(SP)提供偏移地址，将堆栈段寄存器的内容左移 4 位后和偏移地址相加，即获得堆栈操作的物理地址。

④ 附加段寄存器(ES)：用于字符串操作。当进行字符串操作时，附加段作为目的区地址使用，ES 存放附加段的段地址，DI 存放目的区的偏移地址。

分段寻址的好处：把程序区、堆栈区和数据区互相隔离开来，使不同程序在不同区域中运行时，可以共享一个数据区。

3. 标志寄存器

8086/8088 内部有一个 16 位的标志寄存器 FLAGS，设置了 9 位标志。其中 3 位是控制标志，它是用专门的置 1 或清除指令人为设置的，以此来控制 CPU 的操作；6 位是状态标志，它们反映了执行操作后算术或逻辑运算结果的特征，8086/8088 可以根据这些标志位的状态决定其后续动作。剩下的 7 位未用。各标志位的安排如图 2.2-4 所示。



图 2.2-4 标志寄存器

6 位状态标志为：CF、PF、AF、ZF、SF、OF。3 个控制标志为：DF、IF、TF。

① CF：进位标志。在进行算术运算时，最高位产生进位或借位时使 CF 置 1；否则 CF 置 0。移位指令也影响这一标志。还可用有关指令置位、清零或取反。

② PF：奇偶标志。若操作结果中低 8 位“1”的个数为偶数，PF = 1；否则 PF = 0。

③ AF：辅助进位标志。在进行字节运算时，由低半字节向高半字节产生进位或借位时，AF = 1；否则为 0。AF 与 CF 一起，用来对 BCD 码运算的结果进行十进制调整。

④ ZF：零标志。当前运算结果为零时，ZF = 1；否则 ZF = 0。

⑤ SF：符号标志。与运算结果的最高位相同，结果为负数 SF = 1；否则 SF = 0。

⑥ OF：溢出标志。所谓溢出是指在算术运算中，带符号数的运算结果超出了所能表达的范围。例如，字节的运算结果超出了-128~127 范围，或字的运算结果超出了-32768~32767 的范围。溢出时 OF = 1，否则 OF = 0。

⑦ DF：方向标志，用于在字符串操作中规定数据处理的方向。在 DF = 1 时，字符串操作指令修改地址指针用减法，此时字符串处理从高地址向低地址进行。若 DF = 0，则相反，字符串操作指令修改地址指针用加法，即字符串处理从低地址向高地址进行。

⑧ IF：中断允许标志。IF = 1 时允许 CPU 响应可屏蔽中断；若 IF = 0，禁止 CPU 接受外界的可屏蔽中断请求。该标志可用有关指令置位或清零，从而控制 CPU 是否响应可屏蔽中断。

⑨ TF：跟踪标志。当 TF = 1 时为单步操作，CPU 每执行一条指令后进入内部中断，以便对指令的执行情况进行检查；若 TF = 0，则 CPU 处于正常的连续执行指令状态。

【例 2.2-1】 两个八位有符号数 40H 和 41H 执行加法运算后，系统的标志位将如何变化？

【解答】

01000000

+ 01000001

10000001

⋮

⋮

⋮

+64

+ 65

+129

当执行加法运行时，最高位没有向前产生进位，CF=0；操作结果中低 8 位“1”的个数为偶数，PF=1；D₃ 位没有向 D₄ 位产生进位，AF=0；结果不是 0，ZF=0；由于最高位为 1，SF=1；两个数相加的结果超出了字节运算的范围(-128~127)，产生了溢出，OF=1。

值得指出的是 CF 是对无符号数运算有意义的标志位，而 OF 是对有符号数运算有意义的标志位。因此，在 CF/OF 标志位的利用上，编程者必需考虑数据的类型。

2.2.3 8086/8088 存储器寻址

8086/8088 系统中存储器按字节编址，20 条地址线可寻址的存储器空间为 $2^{20} = 1\text{MB}$ ，对应存储器的地址编码为 00000H~FFFFFH。在这一范围内，每一个存储单元(1B)由 20 条地址线的一个唯一的编码与之对应。这个编码也就是存储单元的地址。这种实际的存储器编码地址称为物理地址或绝对地址。

访问一个存储单元必须使用 20 位的地址信息，但 8086/8088 内部的数据总线和寄存器都是 16 位的，无法存储 20 位的物理地址。因此，对地址的传送和运算只能在 16 位范围内进行，也就是说，CPU 的寻址能力被局限在 $2^{16} = 64\text{KB}$ 的区域内。如何解决这一问题呢？可以设想，任何一个 20 位的地址都可分成两部分 16 位地址，例如：8FFFFH=80000H + FFFFH，这样可把 8000H 存储到一个 16 位的寄存器 DS，把 FFFFH 存储到另一个 16 位寄存器 BX，需要物理地址时，用一个 20 位的加法器把 DS 的内容 8000H 乘 16(相当于把 8000H 左移 4 位)，再把 BX 的内容 FFFFH 加在低 16 位上，即形成了 20 位的物理地址。如图 2.2-5 所示。

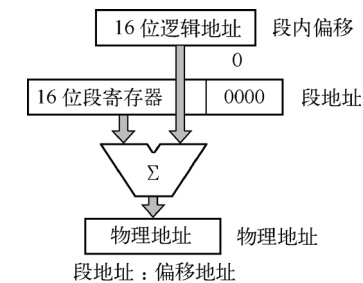


图 2.2-5 8086/8088 物理地址的形成

当 DS 内容不变，BX 的内容从 0000H 变到 FFFFH 时，所产生的地址范围为 80000H~8FFFFH，即 64KB 寻址空间。显而易见，DS 指示了物理地址的高位地址，BX 的内容为这一范围内的相对变化量(偏移量)。DS 和 BX 中的地址都不是绝对地址，而是一个相对的地址，通常称这种程序中编排的地址为逻辑地址，它包括两部分：段地址和偏移地址。用这一方法把 1MB 的存储空间分成若干段，每个段的容量不超过 64KB。把段的起始地址的高 16 位(称段基址)存放到段寄存器，段内的相对地址(相对于段起始地址的偏移值，称偏移地址，又称有效地址)存放到另一些 16 位寄存器中，经过 20 位的加法器就可以形成任意一个 20 位的物理地址。

如果把 1MB 的存储空间划分为连续的固定的 64KB 的逻辑段，可以分成 16 个逻辑段，如图 2.2-6 中所示的 0000~F000 段。在实际应用中，允许将 1MB 存储器分成更多的逻辑段，它们可在整个存储空间中浮动，也就是说，段与段之间可以部分重叠，完全重叠，连续排列，断续

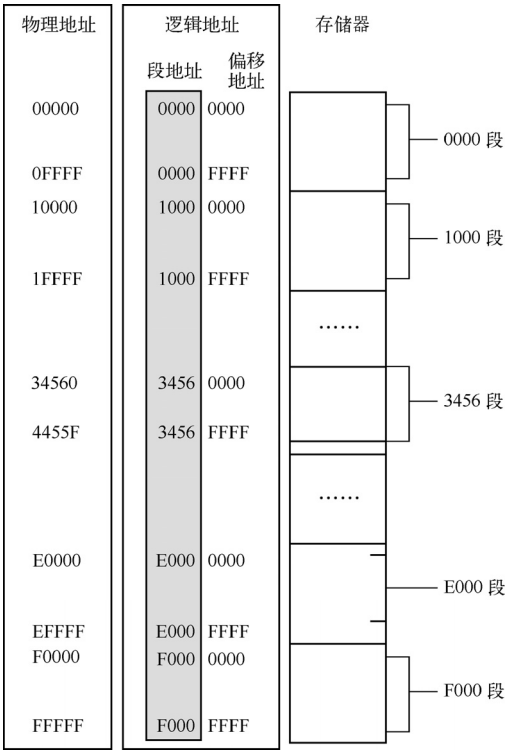


图 2.2-6 存储器段示意图

排列，如图 2.2-6 中所示的 3456 段，其中实际地址 4455FH 可以从两个部分重叠的段中得到：一个段地址为 3456H，偏移地址为 FFFFH；另一个段地址为 4000H，而偏移地址为 4455FH，由此可见，尽管使用了不同的段地址和偏移地址，但它们仍然指向同一个实际地址。从 20 位地址形成的原理可以看出，段基址是 20 位地址中的高 16 位的部分，相当于低 4 位为零的 20 位地址，因此，段的起始地址必须是低 4 位为零的那些地址。

需要强调的是，段区的分配工作是由操作系统完成的，但必要时程序员可以指定所需占用的内存区域。

计算机的存储器中存放着 3 类信息：代码——指令的机器码，指示 CPU 执行什么操作；数据——数值和字符，程序加工的对象；堆栈——临时保存的返回地址和中间结果。为了避免混淆，这 3 类信息通常分别存放在各自的存储器区域内。用段寄存器指示这些存储区域的起始地址。一般每个段中存放着同类性质的信息。

8086/8088 系统通常约定用 4 个 16 位的段寄存器存放上述信息的段基址。这 4 个段寄存器分别是：CS—16 位代码段寄存器；DS—16 位数据段寄存器；ES—16 位附加段(第二数据段)寄存器；SS—16 位堆栈段寄存器。

【例 2.2-2】 数据段内一个数据的偏移地址 EA = 5678H，且 DS = 1234H，则该数据在内存中的物理地址是多少？该数据段首单元和末单元的物理地址是多少？

【解答】 物理地址 = 1234H × 16 (左移 4 位) + 5678H = 17AC8H
首单元的物理地址 = 1234H × 16 + 0000H = 12340H
末单元的物理地址 = 1234H × 16 + FFFFH = 2233FH

2.2.4 8086/8088 外部引脚

8086 微处理器采用双列直插式封装结构，共有 40 个引脚，如图 2.2-7 所示。它的外部引脚由以下几类信号组成：



图 2.2-7 8086/8088 芯片引脚

- (1) 为微处理器芯片的工作提供电源、时钟、复位等必备的信号；
- (2) 与存储器之间交换数据时所需要的地址线、数据线和存储器读写控制线；

(3) 与 I/O 端口之间交换数据所需要的地址线、数据线、I/O 读写控制线、I/O 接口的中断管理、DMA 控制线及时钟复位信号等。

当用 8086 微处理器构成一个微机系统时，8086 有两种不同的组态或模式可供选择，即最大组态和最小组态。选择哪种工作模式，主要取决于应用系统的规模。

当用 8086 微处理器构成一个单 CPU 系统时，存储器组织芯片较少，I/O 端口较少，如单板机规模。这种规模组成的系统，只需要加上少量的锁存器和缓冲器就可隔离它的地址、数据和控制总线，无须太多驱动电路，读写控制可直接由 CPU 来控制管理。在这种情况下，一般选择 8086 最小工作组态，它既满足设计需要，又减少了辅助电路(见图 2.3-1)。

若要构成较大的系统，如存储器容量大、I/O 接口多的情况；或者两片以上的微处理器系统，如微型计算机系统的规模。外部系统总线不总是由一个微处理器控制，而是由两个以上的微处理器分时控制，为此，在系统中设置一个总线控制器，由控制着系统的微处理器发送信号给总线控制器，总线控制器发出控制总线的信号。这种系统组织就要选择 8086 的最大工作组态(见图 2.3-3)。

用一个引脚 $\overline{MN}/\overline{MX}$ 来规定 8086 处在什么工作模式。若 $\overline{MN}/\overline{MX}$ 引脚连至电源(+5V)，则是最小组态；若把它接地，则 8086 处在最大工作模式。这两种组态下，8086/8088 引脚中的第 24~31 脚的名称和意义是不同的，如图 2.2-7 所示，要注意其意义与区别。

按照微处理器引脚的用途归类可分为地址线、数据线和控制线，下面详细说明 8086/8088 引脚的功能及使用方法。

1. 地址/数据总线(AD₁₅~AD₀，三态，双向)

这是一组地址与数据时分复用信号线，所谓时分复用，是指一条信号线要传输一种以上的信号，它是从信号占用的时间来加以区分的。地址/数据线在第一个时间段作为地址信号线，在第二个时间段作为数据信号线，它们都是并行的 16 位输入或输出。具体实现时是这样的：当 ALE 为高电平时，该组信号做地址线；当 ALE 信号为低电平时，该组信号做数据线。8088 外部数据总线只有 8 条，所以时分复用的地址/数据总线为 AD₇~AD₀。

使用时，当 CPU 访问存储器或对输入/输出端口操作时，都要复用这组信号线，下面的总线时序分析中会详细解释如何复用。

2. 地址/状态总线(A₁₉/S₆、A₁₈/S₅、A₁₇/S₄、A₁₆/S₃，输出，三态)

A₁₉~A₁₆ 是地址信号的高 4 位，和 AD₁₅~AD₀ 一样，也是时分复用的信号，当 ALE 为高电平时，该组信号做地址线，当 ALE 信号为低电平时，该组信号做状态线。地址线和状态线都是由 CPU 发送出来的，所以它们都是输出信号线。

4 位状态线表示 CPU 当前工作状态，有不同用途。

表 2.2-1 S₄、S₃ 状态

S ₄	S ₃	段寄存器
0	0	ES
0	1	SS
1	0	CS
1	1	DS

(1) S₄、S₃ 用来指示当前正在使用哪个寄存器，如表 2.2-1 所示。

(2) S₅ 用来指示中断允许标志 IF 的状态，S₅ = 1，表示当前允许可屏蔽中断请求；S₅ = 0，则禁止一切可屏蔽中断。

(3) S₆ 始终为 0，用来指示 8086/8088 当前与总线相连。

3. 控制总线

最大和最小组态下都要使用的公共信号线如下。

(1) $\overline{BHE}/S_7$ (输出，三态)——高 8 位数据总线允许/状态复用引脚，低电平有效。8086 在总线周期的 T₁ 状态用 $\overline{BHE}$ 指出高 8 位数据总线上的数据有效。在 T₂~T₄ 时，S₇ 输出状态信息，在 8086 芯片设计中，S₇ 未赋予实际意义，始终为逻辑 1。

(2) $\overline{MN}/\overline{MX}$ ——最大/最小模式控制引脚。若 $\overline{MN}/\overline{MX}$ 引脚连至电源(+5V)，则为最小组态；接地，则 8086 处在最大组态。

(3) $\overline{RD}$ (输出，三态)——读选通信号，低电平有效。当其有效时，表示正在进行存储器读或 I/O 读。在 DMA 方式时，此线浮空。

(4) $\overline{READY}$ (输入)——准备就绪信号。这是所寻址的存储器或 I/O 设备发给 CPU 的响应信号，高电平有效。当其有效时，将完成数据传送。CPU 在 T_3 周期开始采样 $\overline{READY}$ 线，若其为低电平，则在 T_3 周期结束以后插入 T_w 周期，直至 $\overline{READY}$ 变为有效，则在此 T_w 周期结束以后，进入 T_4 周期，完成数据传送。

(5) $\overline{INTR}$ (输入)——可屏蔽中断请求信号。它是一个电平触发输入信号，高电平有效。CPU 在每一个指令周期的最后一个 T 状态采样这条线，以决定是否进入中断响应周期。这条线上的请求信号，可以用软件复位内部的中断允许标志位来加以屏蔽。

(6) $\overline{TEST}$ (输入)——这个检测输入信号是由“Wait”指令来检查的。若此输入脚有效(低电平有效)，则执行继续；否则处理器就等待，进入空转状态。这个信号在每一个时钟周期的上升沿由内部同步。

(7) $\overline{NMI}$ (输入)——非屏蔽中断信号，是一个边沿触发信号。这条线上的中断请求信号不能用软件来加以屏蔽，所以当这条线由低到高变化时，就在现行指令结束以后引起中断。

(8) $\overline{RESET}$ (输入)——复位请求信号，该信号有效，将使 CPU 回到其初始状态；当其无效时，CPU 将重新开始工作。为保证可靠复位，在上电复位(冷启动)时，要求 $\overline{RESET}$ 的有效时间应维持 50 μ s 以上；在不掉电复位(热启动)时，该信号的有效时间应维持在 4 个时钟周期以上。

(9) CLK (输入)——时钟输入信号。它提供了处理器和总线控制器的定时操作。8086 的标准时钟频率为 5MHz。

4. 电源线和地线

电源线 V_{CC} 接入电压为 $5V \pm 10\%$ 。地线 GND 接地。

5. 其他信号线

8086/8088 CPU 的第 24~31 引脚是一些控制信号线，它们根据 $\overline{MN}/\overline{MX}$ 信号连接方式决定的 8086/8088 工作模式不同，其名称和意义会有很大区别，在 2.3 节将详细介绍。

练习题 1

2.2-1 8086 CPU 在结构上由两个独立的处理单元_____和_____构成。

2.2-2 将 62A0H 和 4321H 相加，则 AF = __, SF = __, ZF = __, CF = __, OF = __, PF = __。

2.2-3 设某台微处理器有 20 根地址总线，16 根数据总线，若存储器寻址空间和 I/O 地址的寻址空间是分开的，则存储器空间大小为____，经数据线一次可传送的带符号数的范围是_____。

2.2-4 已知 CS=1800H, IP=1500H, 则指令所处的物理地址为_____, 给定一个数据的有效地址是 2359H, 且 DS = 49B0H, 该数据在内存中的实际物理地址为_____H。

2.2-5 8086/8088 段寄存器的功能是 ()。

A. 用于计算有效地址 B. 执行各种数据传送操作 C. 用于存放段起始地址及计算物理地址

2.2-6 微机的地址总线功能是 ()。

A. 只用于选择存储器单元 B. 只用于选择进行信息传输的设备
C. 用于传送要访问的存储器单元或 I/O 端口的地址 D. 只用于选择 I/O 端口

2.2-7 在堆栈段中，存放栈顶地址的寄存器是 ()。

A. IP B. SP C. BX D. BP

2.3 8086/8088 CPU 工作模式和引脚

2.3.1 8086/8088 最小工作模式

1. 最小工作模式引脚

当 $\overline{MN}/\overline{MX}$ 引脚接+5V 时, CPU 处于最小工作模式, 第 24~31 引脚功能和使用方法如下:

(1) $\overline{M}/\overline{IO}$ (输出, 三态)——这条引脚用以区分是存储器访问还是 I/O 访问。若此引脚为高电平, 则为存储器访问; 若此引脚为低电平, 则为 I/O 访问。当 CPU 处在 DMA 响应时, 此脚浮空。8088 此引脚为 $\overline{IO}/\overline{M}$ 。

(2) $\overline{WR}$ (输出, 三态)——此脚低电平有效, 是 CPU 写操作时输出的一个选通信号, 表示是处在存储器写或 I/O 写(取决于 $\overline{M}/\overline{IO}$ 信号)周期。此信号在整个 T_2 、 T_3 、 T_w 状态有效。当 CPU 处在 DMA 响应时此脚浮空。

(3) $\overline{INTA}$ (输出)——是 CPU 输出的中断响应信号, 低电平有效。它在每一个中断响应周期的 T_2 、 T_3 和 T_w 状态有效, 可用做中断向量选通信号。

(4) ALE(输出)——地址锁存允许信号, 高电平有效, 这是一个在 T_1 状态时钟低时由 CPU 提供的正选通脉冲, 把在地址-数据线 $AD_{15}\sim AD_0$ 和地址-状态线 $A_{19}\sim A_{16}$ 上出现的地址信号, 锁存到用三态输出锁存器 74LS373 组成的地址锁存器中去。

(5) $\overline{DT}/\overline{R}$ (输出, 三态)——数据发送/接收信号(Date Transmit/Receiver)。在最小系统组态时, 若为了增强数据总线的驱动能力, 用双向三态数据缓冲器 74LS245 实现数据的发送和接收。74LS245 的方向控制线就用这个信号控制, 以确定数据传送的方向。此引脚为高电平, 则数据发送(CPU 写); 为低, 则为接收(CPU 读)。当 CPU 处在 DMA 响应时, 此脚浮空。

(6) $\overline{DEN}$ (输出, 三态)——数据允许(Date Enable)信号, 低电平有效。该信号连接在数据双向缓冲器 74LS245 的输出使能信号上, 作为 74LS245 芯片的输出允许信号。它在每一次存储器访问、I/O 访问或中断响应周期有效。当 CPU 处在 DMA 响应时, 此脚浮空。

(7) HOLD(输入), HLDA(输出)——HOLD 是系统中别的总线主设备要求占用总线时, 向 CPU 发出的总线请求信号, 高电平有效。当 CPU 接收到有效的 HOLD 信号后, 在当前总线周期的 T_4 状态输出一个高电平有效的总线请求响应信号(HOLD Acknowledge)HLDA。同时, CPU 就使地址线、数据线和相应的控制信号线浮空。当 CPU 检测到 HOLD 信号变为低电平后, 它使 HLDA 变为低电平, 同时又控制总线。

注意: 在 8088 中, 没有 $\overline{BHE}/S_7$ 引脚, 对应的引脚定义为状态信号 $\overline{SS}_0$, 它与 $\overline{IO}/\overline{M}$ 线和 $\overline{DT}/\overline{R}$ 线一起, 反映现行总线周期的状态, 如表 2.3-1 所示。

2. 最小工作模式系统结构

在最小工作模式下, $\overline{M}/\overline{IO}$ 、 $\overline{WR}$ 和 $\overline{RD}$ 组合决定 CPU 是访问存储器还是执行输入/输出操作。图 2.3-1 示出了一种典型的 8086 最小组态方式的系统结构。

表 2.3-1 $\overline{SS}_0$ 系统状态

$\overline{IO}/\overline{M}$	$\overline{DT}/\overline{R}$	$\overline{SS}_0$	性能
1	0	0	中断响应
1	0	1	读 I/O 端口
1	1	0	写 I/O 端口
1	1	1	暂停(Halt)
0	0	0	取指
0	0	1	读存储器
0	1	0	写存储器
0	1	1	无源

图 2.3-1 中 74LS373 是 8 位三态输出锁存器，其输入使能端 LE 是锁存控制信号，连接到 CPU 的地址锁存信号 ALE 上，由 ALE 信号对其进行锁存控制，当 ALE 信号有效时，表示在地址锁存时段，数据/地址复用信号线 AD₁₅~AD₀ 用做地址线，地址被锁存在 74LS373 的输出端，该锁存着的地址就是访问存储器单元地址或操作输入/输出端口地址。当 ALE 为低时，表示进入数据交换时段，数据/地址复用信号线 AD₁₅~AD₀ 用做数据线。

74LS245 是 8 位三态双向数据缓冲器，DIR 信号线控制输入/输出的方向，74LS245 的 $\overline{G}$ 信号是输出使能信号。

当 CPU 发出写信号时，表示欲将 CPU 中的某个寄存器内容输出到存储器或 I/O 输出端口；当 $\overline{DEN}$ 有效，DT/ $\overline{R}$ 为高电平时，使 74LS245 双向数据缓冲器实现由 CPU 写到存储器或 CPU 输出到 I/O 端口的数据流。

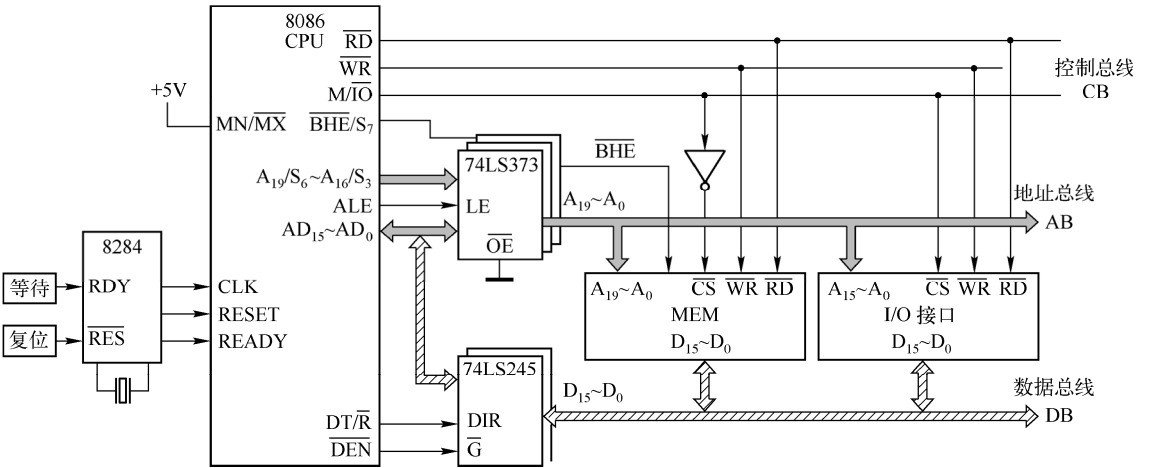


图 2.3-1 8086 CPU 最小组态方式系统结构

当 CPU 发出读信号时，表示将存储器或 I/O 输入端口的数据传输到 CPU 的某个寄存器中；当 $\overline{DEN}$ 有效，DT/ $\overline{R}$ 为低电平时，使 74LS245 双向数据缓冲器实现由存储器到 CPU 或 I/O 端口到 CPU 的数据流。

2.3.2 8086/8088 最大工作模式

1. 最大工作模式引脚

当 MN/ $\overline{MX}$ 引脚接地时，8086/8088 微处理器处于最大组态工作模式下，它的第 24~31 引脚与最小组态意义不同，这 8 条引脚的功能和使用方法如下：

(1) $\overline{S}_2$ 、 $\overline{S}_1$ 、 $\overline{S}_0$ (输出，三态)——状态线的编码如表 2.3-2 所示。

8288 总线控制器根据 8086 微处理器发出的 $\overline{S}_2$ $\overline{S}_1$ $\overline{S}_0$ 信号的编码，进入表 2.3-2 所示的总线周期，用于控制有关存储器访问或 I/O 操作的总线周期，并在每个总线周期产生各种所需要的控制信号。

在时钟周期 T_1 状态期间， $\overline{S}_2$ $\overline{S}_1$ $\overline{S}_0$ 的任一组合，指示一个总线周期的开始；而在 T_3 或 T_w

表 2.3-2 8086 发出总线控制编码

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	性能
0	0	0	中断响应
0	0	1	读 I/O 端口
0	1	0	写 I/O 端口
0	1	1	Halt
1	0	0	取指
1	0	1	读存储器
1	1	0	写存储器
1	1	1	过渡状态

期间返回到过渡状态(111)，即表示一个总线周期的结束。

当 CPU 处在 DMA 响应状态时，这些引脚浮空。

(2) $\overline{RQ}/\overline{GT}_0, \overline{RQ}/\overline{GT}_1$ (输入/输出)——请求/允许(Request/Grant)脚，是由外部的总线主设备请求总线并促使 CPU 在现行总线周期结束后让出总线用的。每一个引脚都是双向的， $\overline{RQ}/\overline{GT}_0$ 比 $\overline{RQ}/\overline{GT}_1$ 有更高的优先权。这些引脚的内部有一个上拉电阻，所以允许这些引脚浮空。请求和允许的顺序如下：

① 由其他的总线主设备输送一个宽度为一个时钟周期的脉冲给 8086/8088，表示总线请求，相当于 HOLD 信号。

② CPU 在当前总线周期的 T_4 或下一个总线周期的 T_1 状态，输出一个宽度为一个时钟周期的脉冲给请求总线的设备，作为总线响应信号(相当于 HLDA 信号)，从下一个时钟周期开始，CPU 释放总线。

③ 当外设的 DMA 传送结束时，总线请求主设备输出一个宽度为一个时钟周期的脉冲给 CPU，表示总线请求的结束。于是 CPU 在下一个时钟周期又开始控制总线。

每一次总线主设备的改变，都需要这样的 3 个脉冲(低电平有效)。在两次总线请求之间，至少要有一个空时钟周期。

(3) $\overline{LOCK}$ (输出，三态)——低电平有效，当其有效时，其他总线主设备不能获得对系统总线的控制。 $\overline{LOCK}$ 信号由前缀指令“LOCK”使其有效，且在下一个指令完成以前保持有效。当 CPU 处在 DMA 响应状态时，此引脚浮空。

(4) QS_1, QS_0 (输出)—— QS_1 和 QS_0 提供一种状态 (Queue Status)，允许外部追踪 8086/8088 内部的指令队列，如表 2.3-3 所示。队列状态在 CLK 周期期间是有效的，在这以后，队列的操作已完成。

表 2.3-3 $QS_1 QS_0$ 状态

QS_1	QS_0	性 能
0	0	无操作
0	1	从指令队列取走第一个字节
1	0	队列空
1	1	从指令队列取走后继字节

2. 总线控制器

最大工作模式系统中，会出现两个以上的微处理器，在某个时刻，只有一个微处理器管理着总线，为了协调微处理器对总线管理的更替，使用总线仲裁器进行仲裁。一旦仲裁之后，就确立了由哪个微处理器管理总线。该微处理器就向总线控制器发送 $\overline{S}_2 \overline{S}_1 \overline{S}_0$ 控制编码。

8288 就是为解决这一问题而设计的，它为专用于 8086/8088 微处理器系列最大模式系统中的总线控制器。在最大模式下，微处理器发出控制编码，8288 总线控制器按时序要求发出控制信号，不再由 8086 直接输出。8288 的引脚和内部结构如图 2.3-2 所示。

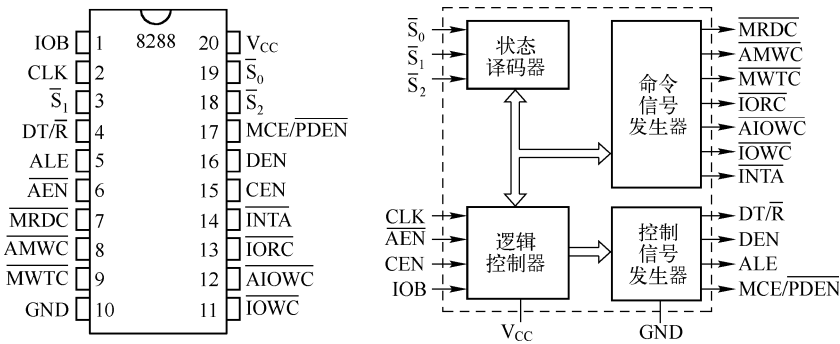


图 2.3-2 8288 总线控制器引脚与内部结构

(1) 8086/8088 的控制编码 $\overline{S_2} \overline{S_1} \overline{S_0}$ 与 8288 输出命令关系

8288 总线控制器接收 8086/8088 的控制编码 $\overline{S_2} \overline{S_1} \overline{S_0}$ ，经 8288 状态译码后，确定系统总线执行何种操作，每个控制编码只能使表 2.3-4 中的某一行有效。

(2) 8288 输出信号

8288 输出信号线是多选一的，每个输出与微处理器的某个总线周期相对应，因此，下面又将选中某条信号线称做执行某个命令。

① $\overline{MRDC}$ (输出)——存储器读命令，相当于最小模式系统中由 CPU 发出的控制信号 $\overline{RD}$ 和 $M/\overline{IO}$ 为高电平时的组合。指示存储器把指定存储单元的内容送到数据总线上。

② $\overline{IORC}$ (输出)——I/O 端口读命令，相当于最小模式中由 CPU 发出的控制信号 $\overline{RD}$ 和 $M/\overline{IO}$ 为低电平时的组合。指示 I/O 端口把数据送到数据总线上。

③ $\overline{MWTC}$ (输出)——存储器写命令，相当于最小模式系统中由 CPU 发出的 $\overline{WR}$ 和 $M/\overline{IO}$ 为高电平时的组合。该信号指示存储器接收总线上的数据，并将数据写入到指定的存储单元。

④ $\overline{IOWC}$ (输出)——I/O 端口写命令，相当于最小模式系统中的控制信号 $\overline{WR}$ 和 $M/\overline{IO}$ 为低电平时的组合。该信号指示 I/O 端口接收总线上的数据，并将数据写入到指定的 I/O 端口。

⑤ $\overline{AMWC}$ 和 $\overline{AIOWC}$ (输出)——存储器和 I/O 端口超前写命令，这两个信号的作用同 $\overline{MWTC}$ 和 $\overline{IOWC}$ 。其区别是 $\overline{AMWC}$ 和 $\overline{AIOWC}$ 分别比 $\overline{MWTC}$ 和 $\overline{IOWC}$ 提前一个时钟周期出现。当系统配备的 I/O 设备和存储器速度较慢时，使用这两个信号就能得到一个额外的时钟周期执行写入操作。

⑥ $\overline{INTA}$ (输出)——中断响应信号，与最小模式下由 CPU 直接发出中断响应信号相同。

(3) 8288 的控制信号

① CEN(输入)——命令允许信号，由外部输入，高电平有效。当有多片 8288 协同工作时起片选作用。CEN 有效时，允许 8288 输出所有类型控制信号。CEN 为低电平时，禁止发控制信号，同时使 $\overline{DEN}$ 和 $\overline{MCE/PDEN}$ 呈高阻态。任何时候只允许 1 片 8288 的 CEN 信号为高电平。

② $\overline{AEN}$ (输入)——地址允许信号，由总线仲裁器输入，低电平有效。用于多总线之间的同步控制。当 $\overline{AEN}$ 变为无效时，8288 的命令输出引脚立即进入高阻状态。只有当 CEN 和 $\overline{AEN}$ 都有效时，8288 才能正常输出命令和控制信号。

③ IOB(输入)——总线方式控制信号，高电平有效。当 IOB = 1 时，8288 工作于 I/O 总线方式，表示 8288 只用来控制 I/O 总线，仅对 I/O 端口进行控制；当 IOB = 0 时，8288 工作于系统总线方式，这时 8288 产生访问存储器和 I/O 端口的全部控制信号。一般情况下，IOB = 0。

④ $\overline{MCE/PDEN}$ (输出)——主控级联/外设数据允许信号，向外部输出。当 8288 工作在系统总线方式时，用做主控级联允许信号 MCE。在中断响应周期的 T_1 状态时 MCE 有效，控制主 8259A 向从 8259A 输出级联地址；工作于 I/O 总线方式时，用做外设数据允许信号 $\overline{PDEN}$ ，用来控制外部设备通过 I/O 总线传送数据。

⑤ $\overline{DEN}$ ——数据传送允许信号，输出，高电平有效。

表 2.3-4 $\overline{S_2} \overline{S_1} \overline{S_0}$ 与 8288 输出对应关系

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	8086 总线周期	8288 命令输出
0	0	0	中断响应	$\overline{INTA}$
0	0	1	读 I/O 端口	$\overline{IORC}$
0	1	0	写 I/O 端口	$\overline{IOWC} \quad \overline{AIOWC}$
0	1	1	暂 停	—
1	0	0	取指令码	$\overline{MRDC}$
1	0	1	读存储器	$\overline{MRDC}$
1	1	0	写存储器	$\overline{MWTC}, \quad \overline{AMWC}$
1	1	1	过渡状态	—

3. 最大工作模式系统结构

图 2.3-3 所示为一种典型 8086 最大组态模式下的系统结构。图中有 2 个微处理器：主微处理器 8086 和协处理器 8087，1 个 8288 总线控制器，它还包括如下的器件：总线地址锁存器 74LS373、总线数据缓冲器 74LS245、存储器系统、输入/输出接口。

其在结构上与最小组态系统的主要区别是增设了 1 个总线控制器 8288 和协处理器, 它们可以构成以 8086 CPU 为核心的多处理器系统。两个处理器的总线输出状态信号 $\bar{S}_2 \bar{S}_1 \bar{S}_0$ 同时接到 8288 上, 但在某个时刻只有一个处理器工作, 由管理总线的处理器发送 $\bar{S}_2 \bar{S}_1 \bar{S}_0$ 信号。

8087 不是一个可独立工作的处理器，称为协处理器。它也是 40 引脚的器件，它的引脚除 QS_1 、 QS_0 、BUSY 和 INT 之外，其余引脚的意义与 8086 最大组态方式下的相应引脚的意义完全相同。通过用 QS_1 、 QS_0 、BUSY 信号与 8086 主处理器挂钩，确定当前管理系统的处理器，哪个处理器管理阶段，哪个处理器的总线信号就起作用，另一个处理器处相应的总线处于挂起浮空状态，也就不会影响总线信号。关于 8087 协处理器在此不做进一步讨论。

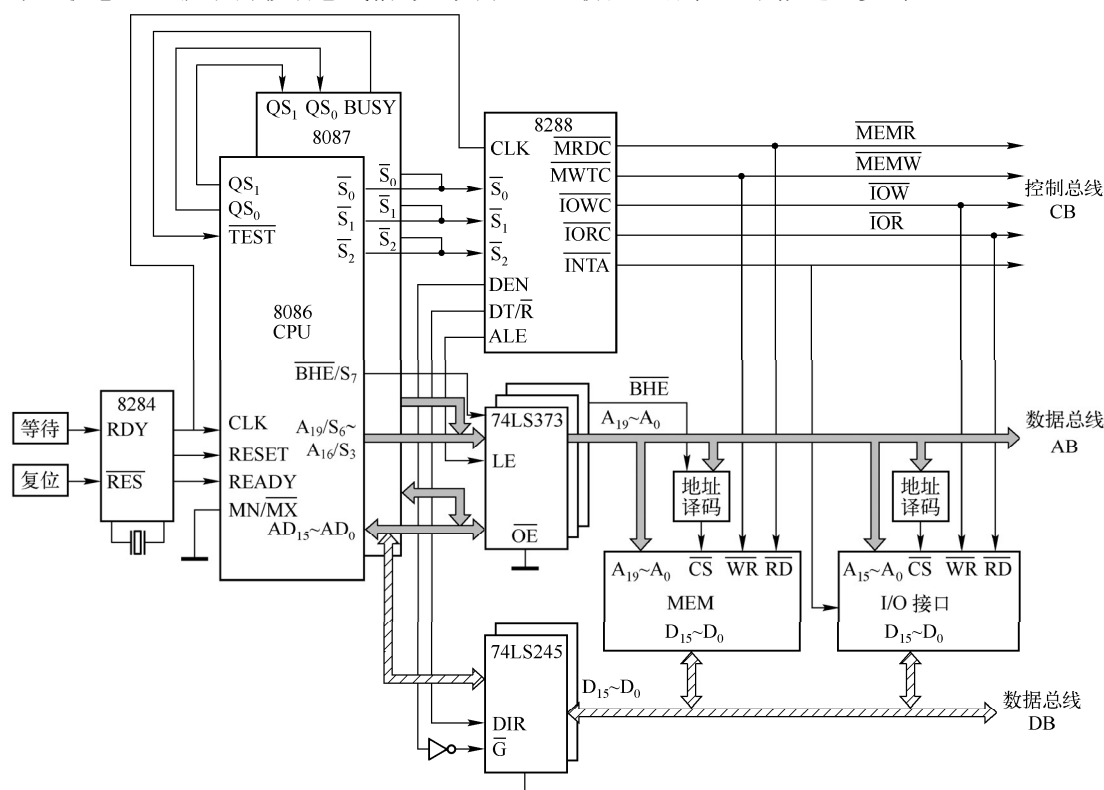


图 2.3-3 8086 CPU 最大组态系统结构

练习题 2

2.3-1 8086 中地址/数据线分时复用, 为保证总线周期内地址稳定, 应配置_____, 为提高总线驱动能力, 应配置_____。

2.3-2 8086 最小组态下的总线控制信号由_____产生, 最大组态下由_____产生。

2.3-3 8086 有两种工作方式, 当 8086 处于最小方式时, $\text{MN}/\overline{\text{MX}}$ 接_____。

- A. +12V B. -12V C. +5V D. 地

2.3-4 8086/8088 CPU 上 INTR 信号为 () 信号有效。

A. 上升沿 B. 下降沿 C. 高电平 D. 低电平

2.3-5 下列说法中属于 8086 CPU 最小工作模式特点的是 ()。

- A. CPU 提供全部的控制信号 B. 由编程进行模式设定
C. 不需要地址锁存器 D. 需要总线控制器 8288

2.4 8086 的总线时序

时序(Timing)是指信号高低电平(有效或无效)变化及相互间的时间顺序关系,总线时序用于描述 CPU 引脚如何实现总线操作。CPU 的时序决定系统各部件间的同步和定时。

微机系统的所有操作都按统一的时钟节拍来进行,这就是 CPU 的系统时钟信号 CLK。8086 的最高时钟频率为 5 MHz。IBM PC/XT 采用时钟发生器 8284 向 CPU 提供 4.77MHz 的时钟信号。每个时钟周期约 210 ns。时钟周期的时间长度就是时钟频率的倒数。

8086 的各种对外操作都需要通过总线,它们都可被称为“总线操作”。总线操作主要有:存储器读和 I/O 读操作、存储器写和 I/O 写操作、中断响应操作、总线请求及响应操作等;还有 CPU 正在进行内部操作,并没有进行实际对外操作的空闲状态。CPU 通过总线操作与外界(存储器和 I/O 端口)进行一次数据交换的过程(时间)称为总线周期(Bus cycle),也称为机器周期(Machine cycle)。一条指令经取指、译码、读写操作数到执行完成的过程称为指令周期(Instruction cycle)。

8086 的基本总线周期需要 4 个时钟周期,每个时钟周期的总线操作并不相同。为了便于区别和叙述,通常将基本总线周期中的 4 个时钟周期编号为 T_1 、 T_2 、 T_3 和 T_4 。因此总线周期中的时钟周期也被称为“T 状态”。

注意,存储器读包括读取指令代码和读取存储器操作数,因为它们都是从存储器中读取信息,存储器读、写和 I/O 读、写是任何一个微处理器最基本和最频繁的 4 种总线周期。

不同工作模式的具体时序有所区分,下面分别介绍。

2.4.1 最小工作模式典型时序

最小组态的典型时序就是存储器读写和 I/O 读写时的总线周期,学习总线周期可以帮助读者了解 CPU 是如何工作的,对用 8086 CPU 组成微型机系统的电路理解与设计将起到作用。

1. 存储器读周期和存储器写周期

存储器的读写总线周期由 4 个时钟节拍 $T_1 \sim T_4$ 组成。下面结合图 2.3-1 的最小组态电路和图 2.4-1、图 2.4-2 的时序图来分析。

(1) T_1 状态

① CPU 根据执行的是访问存储器还是访问 I/O 端口的指令,首先在 $\overline{M}/\overline{IO}$ 线上发出有效电平。若为高电平,表示从存储器读;若为低电平,则表示从 I/O 端口读。此信号将持续整个周期。

② 从地址/数据复用线 $AD_{15} \sim AD_0$ 和地址/状态复用线 $A_{19}/S_6 \sim A_{16}/S_3$ 发要访问的存储器单元的地址(20 位)或发 I/O 端口地址(16 位)。因为这类信号都是复用线,只持续 T_1 状态,因此必须将其锁存起来,以供整个总线周期使用。

③ 为了实现对存储体的高位字节(奇地址存储体)进行寻址,CPU 在 T_1 状态通过 $\overline{BHE}/S_7$ 引脚发 $\overline{BHE}$ 信号。该信号也是复用线,因此也需要锁存。

④ 为了锁存上述信号,CPU 在 T_1 状态从 ALE 引脚上输出一个正脉冲加到 3 片地址锁存器

74LS373 的输入使能端 LE 作为锁存信号。当 ALE 为正时，74LS373 锁存输出随着输入变化(输出等于输入)，此时，地址信号输出到 74LS373 的输出端。当 ALE 变为低电平时，其输出锁存保持不变，即把 20 位地址和 BHE 锁存到地址锁存器中，并输出到系统的地址总线上，如图 2.4-1 所示。

⑤ 为了控制数据传输方向，使 DT/ $\overline{R}$ 输出低电平，控制 74LS245 的传输方向为存储器到 CPU。DT/ $\overline{R}$ 信号的有效电平一直保持到整个总线周期结束。

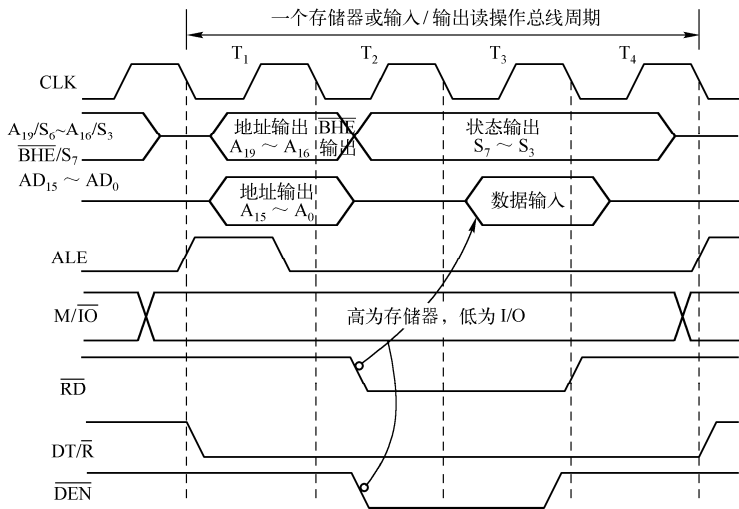


图 2.4-1 存储器读周期时序

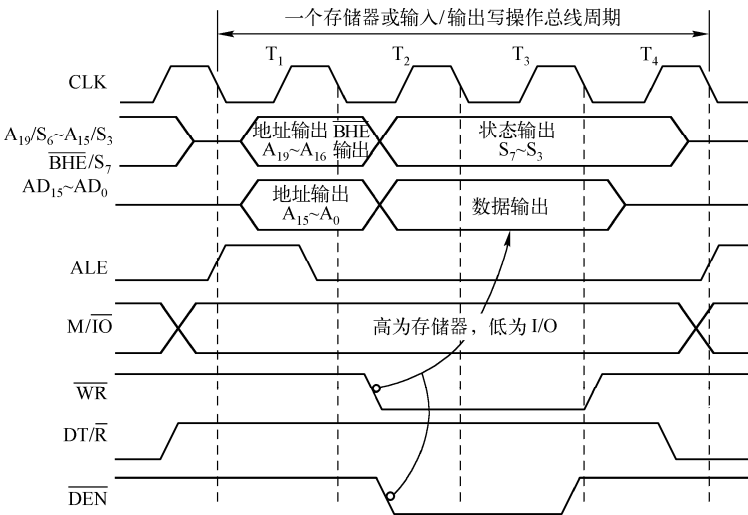


图 2.4-2 存储器写周期时序

(2) T₂ 状态

① CPU 撤销输出的地址信息，AD₁₅~AD₀ 成为高阻态，为读写数据做准备。A₁₆/S₃~A₁₉/S₆ 和 BHE/S₇ 引脚上输出状态信息 S₃~S₇，并一直持续到 T₄ 状态。

② $\overline{DEN}$ 信号变为低电平，允许 74LS245 传输数据。

③ $\overline{RD}$ 引脚输出低电平的读信号到控制总线上，此信号接到系统中所有存储器和 I/O 端口，用来打开存储器的数据单元，以便将数据送至数据总线。

④ $\overline{WR}$ 引脚输出低电平的写信号到控制总线上，此信号接到系统中所有存储器和 I/O 端口，用来打开存储器的数据单元，以便把 CPU 输出到数据总线上的数据写入存储器的选中单元。

(3) T_3 状态

经过 T_1 和 T_2 后, 存储单元把数据送至数据总线, 以供 CPU 读取; 或者 CPU 把要写入存储单元的数据送至数据总线。

在基本总线周期的 T_3 状态, 如果被选中的单元来得及把读出数据送到数据总线上, 或者存储器能及时从数据总线上取走写入数据, 则 8086 系统的等待控制逻辑使 READY 信号有效(高电平), 在 T_3 周期开始的下降沿上, 8086 采样 READY 线上输入的信号。如果 READY 为高电平, 则在 T_3 状态结束 T_4 状态开始的时钟脉冲下降沿把数据总线上的数据读入 CPU 或者写入存储单元。

(4) T_4 状态

8086 完成了数据传输, 在此期间恢复各信号线的初态, 准备执行下一个机器周期。

2. I/O 读周期和 I/O 写周期

I/O 的读写总线周期也由 4 个时钟节拍 $T_1 \sim T_4$ 组成。它的总线周期时序与存储器读写总线周期时序是类似的。所有信号在 $T_1 \sim T_4$ 内的变化都相同, 只是 $\overline{\text{M}}/\overline{\text{IO}}$ 线上发出低有效电平, 表示对 I/O 端口读写操作。

3. 等待周期 T_w

等待周期 T_w 是为了匹配慢速的存储器或 I/O 接口器件读写操作而增设的。

如果是慢速存储器或外设, 在 T_3 期间不能把数据读到数据总线上, 则等待控制逻辑使 READY 为低电平, 8086 采样到 READY 信号无效, 就在 T_3 结束后插入一个等待状态 T_w 。在 T_w 开始的下降沿上, 8086 再对 READY 信号采样, 只要 READY 为低电平, 就在本状态结束后插入一个等待状态, 直至采样到 READY 信号为有效的高电平为止。这时表示慢速部件已把数据送上了数据总线, 在最后一个 T_w 和 T_4 交界的下降沿把总线上的数据读入 CPU, 如图 2.4-3 所示。

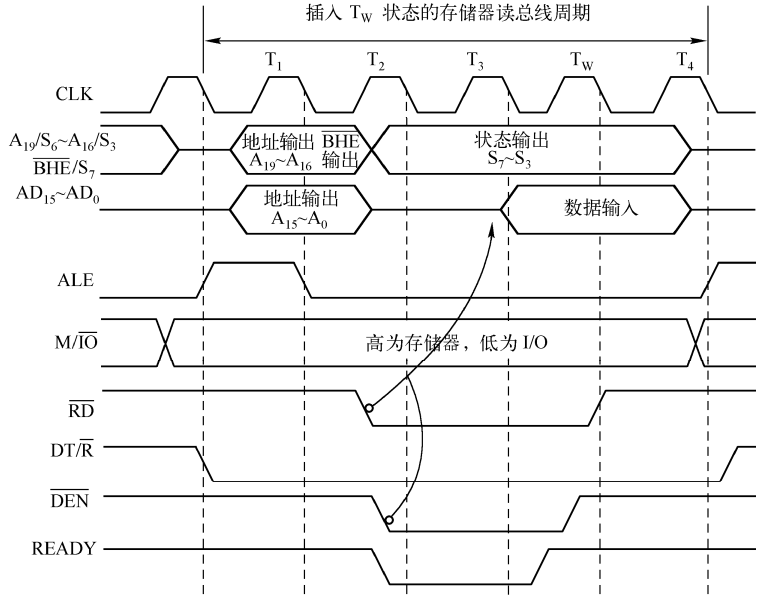


图 2.4-3 具有 T_w 状态的读周期时序

和读周期时序一样, 当存储器或外设端口的工作速度较慢时, 写总线周期也会在 T_3 和 T_4 状态之间插入一个或几个 T_w 状态, 对应的时序图略。

4. 复位周期

8086 的 RESET 信号线，可以用来重新启动系统。

当 CPU 从 RESET 引脚上检测到一个脉冲的上升沿时，便终止当前工作，直到 RESET 信号变为低电平。这时寄存器被复位到初始状态，如表 2.4-1 所示。

8086 要求复位脉冲的有效电平必须至少维持 4 个时钟周期，若为开电源引起的复位，即必须大于 50μs。复位周期的时序如图 2.4-4 所示。

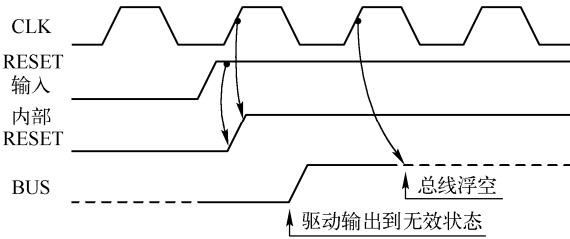


图 2.4-4 复位总线周期时序

表 2.4-1 复位后寄存器的状态

CPU 中的部分	内 容
标志位	清除
指令指针 IP	0000H
CS	FFFFH
DS	0000H
SS	0000H
ES	0000H
指令队列	空

复位时，8086 微处理器使内部寄存器和系统总线分别处于表 2.4-1 和表 2.4-2 所示状态。

表 2.4-2 8086 复位时的总线状态

信 号	状 态
AD ₁₅ ~AD ₀ 、A ₁₉ /S ₆ ~A ₁₆ /S ₃ 、 $\overline{\text{BHE}}/\text{S}_7$ 、M/ $\overline{\text{IO}}$ 、 $\overline{\text{DT}}/\overline{\text{R}}$ 、 $\overline{\text{DEN}}$ 、 $\overline{\text{WR}}$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{INTA}}$	高阻、三态
ALE、HLDA、 $\overline{\text{RQ}}/\overline{\text{GT}}_0$ 、 $\overline{\text{RQ}}/\overline{\text{GT}}_1$ 、QS ₀ 、QS ₁	无效

2.4.2 最大工作模式典型时序

在最大组态下，8086 的基本总线周期常用的有存储器读写和 I/O 读写四个总线周期，除此之外，复位总线周期与最小方式是一致的。

1. 存储器读周期和存储器写周期

在最大组态下，总线的控制信号由 8288 发出，数据和地址总线仍然由 CPU 发出。标准的总线周期也是 4 个时钟周期组成。存储器的读写周期如图 2.4-5 和图 2.4-6 所示。

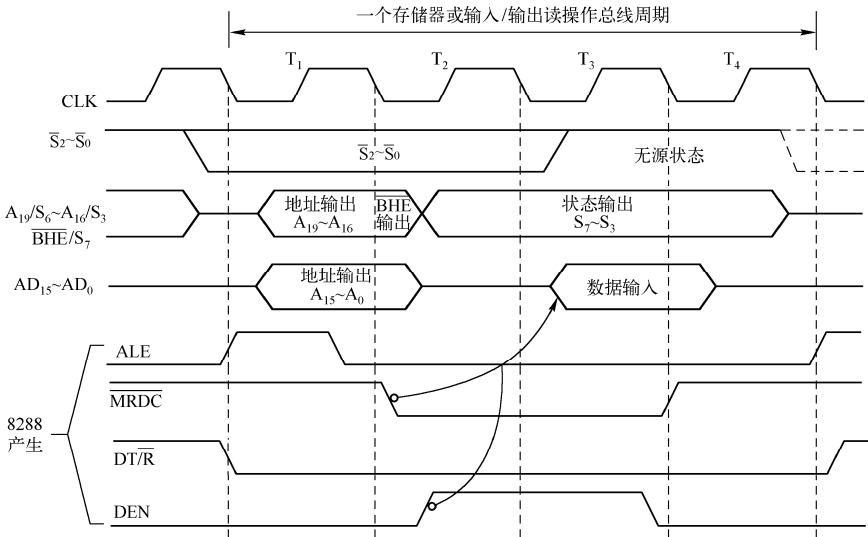


图 2.4-5 最大组态的存储器读周期时序

(1) T₁ 状态

① CPU 发出 20 位地址信号线，同时送出控制编码 $\overline{\text{S}}_2$ 、 $\overline{\text{S}}_1$ 和 $\overline{\text{S}}_0$ 给 8288 总线控制器。8288 对 $\overline{\text{S}}_2$ $\overline{\text{S}}_1$ $\overline{\text{S}}_0$ 进行译码，产生相应的输出命令。